Brick Breaker Proposal

Jemale Lockett and Jonathan Abbott | 6.111 Fall 2013

1 Overview

We want to make an interactive game based upon the classic game brick breaker. The object of brick breaker is to break the bricks that are distributed around the top of the game screen. The bricks get broken after coming in contact with a ball that bounces around the screen. At the bottom is a paddle that in the classic game moves based on user input. The user has to make sure the ball bounces off the paddle without going off the bottom of the screen. In our implementation, instead of using arrow keys to control the paddle's position, we will use a camera to track the position of an actual Ping-Pong paddle held by the user. The control of the game paddle with an actual Ping-Pong paddle will improve the user experience.

2 Design Decisions

We chose the game brick breaker because it is already a fun game that we can improve using an actual Ping-Pong paddle. The actual paddle will make the game more interactive and will allow us to take the users' instinctual movements to allow for a better user experience, in much the same way that current generation video games are employing the technology (e.g. Microsoft Kinect, Nintendo Wii, PlayStation Move). The paddle will have a special colorful lining around its perimeter that will help tracking. A visual representation of the paddle and the camera is shown in Figure 1. It is easiest to track objects by hue, so the algorithum used to determine the location of the paddle will start by identifying the location of the pixels that show the hue of the paddle's perimeter.

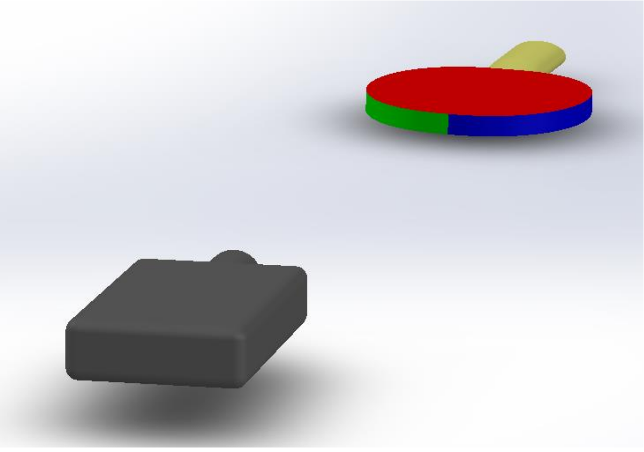

Figure 1: A view of the camera and paddle. The camera will face the paddle that will have a bright strip of two colors on its perimeter. This center points of the two colors will be tracked to determine the location of the paddle for the game.

We will implement a basic set of features with the option to later add extra features to make the game more fun/challenging. The basic game will consist of hitting the ball to direct it towards hitting the bricks above. When the ball comes into contact with a brick, the brick will "break" and disappear. The game will end when all the bricks are cleared or the player has no more balls remaining. The player will start with three balls and lose one ball each time the ball goes past the onscreen paddle and reaches the bottom edge of the display. The player will "win" if all the bricks are broken. A representation of the game screen is given in Figure 2. Figure 2 highlights that one of our goals is to allow the paddle to also tilt. The multicolored ball will be used to help implement "English" or spin on the ball if there is sufficient time at the end.

A special part of the project will be allowing the user to create a level themselves. This will let the user choose the number of bricks and their placement. The user will also be able to specify the bricks' dimensions. The level creation component should prove to be a useful tool in the testing and debugging of the game as well.

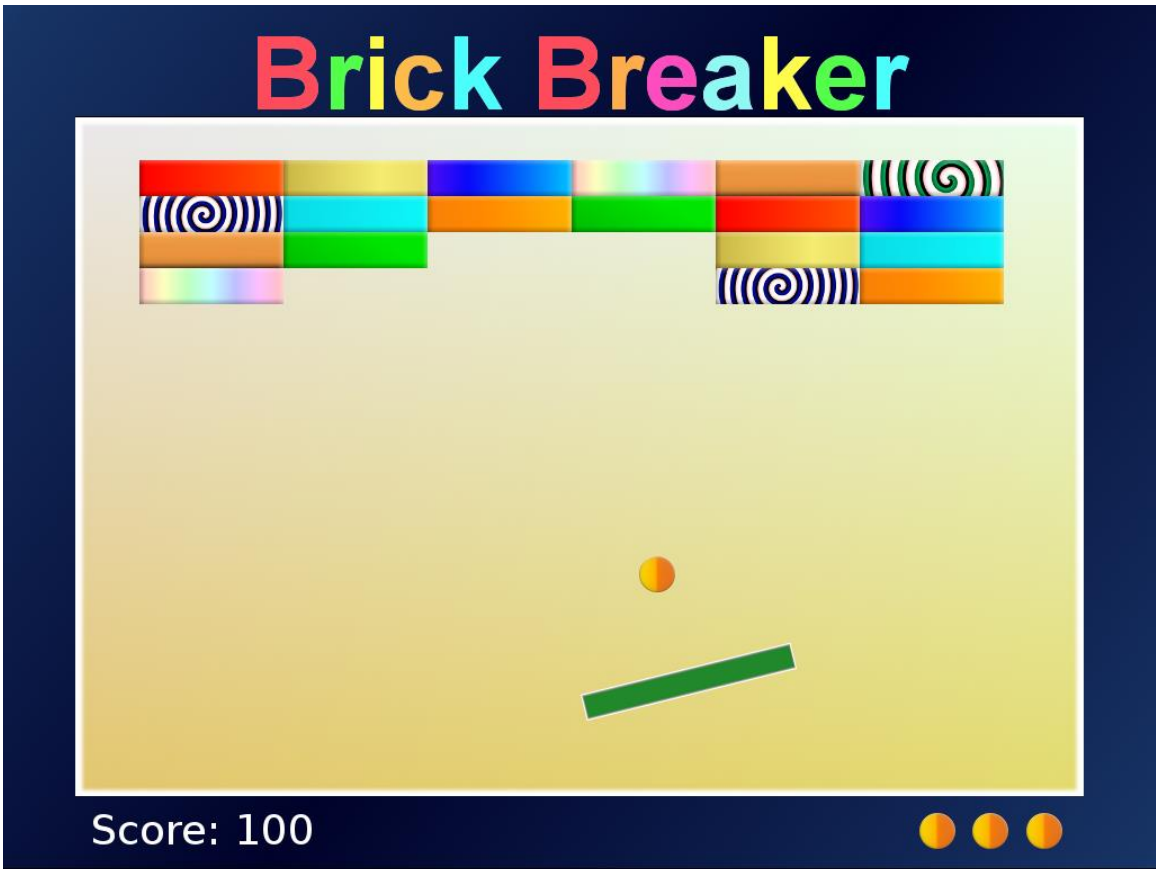

Figure 2: A view of the game screen. This view shows what will be displayed on the monitor: colorful bricks at the top, a paddle that moves and rotates at the bottom, and a two-colored ball. The ball will have two colors to help show spin, which is also a stretch goal for this game. The balls in the right side of the screen represent remaining balls.

3 Implementation

The implementation of brick breaker will be done in four main modules. The modules are to process video input, aid in level creation, perform logic for the game, and to output display and sound from the game. Figure 3 shows how these core components relate. This section describes each of these components in more detail.

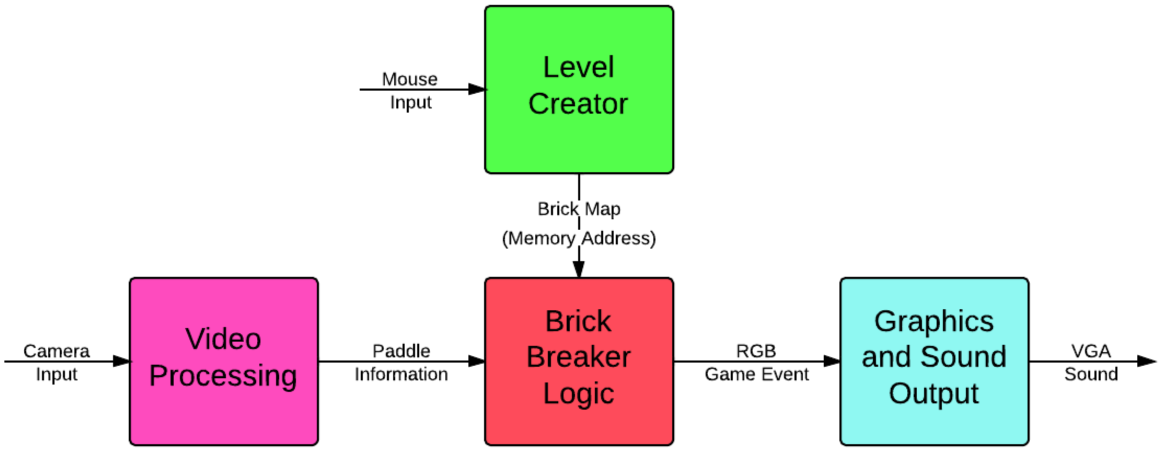

Figure 3: Block diagram of system. The video processing module takes in video and will give information about the paddle to the game logic. The game logic also takes in information about the location of bricks from the level creator module. It uses these two inputs to determine what should be displayed on the screen and what happens in the game. These outputs are sent to the graphics and sound module to be properly formatted and to select the appropriate sound.

3.1 Camera Input Module

The purpose of the camera module is to take in video input and produce output coordinates of the actual Ping-Pong paddle. The module will take in NTSC video data from the camera plugged into the component input jack, decode the NTSC data, create an RGB stream, and finally convert the RGB stream to HSV color space. Converting to HSV allows the module to better identify pixels corresponding to the paddle. The module will determine the central coordinates and the tilt of the paddle. The tilt will be computed by taking the leftmost and rightmost coordinates of the paddle. Jemale will work on this module, a block diagram of which is shown below in Figure 4.

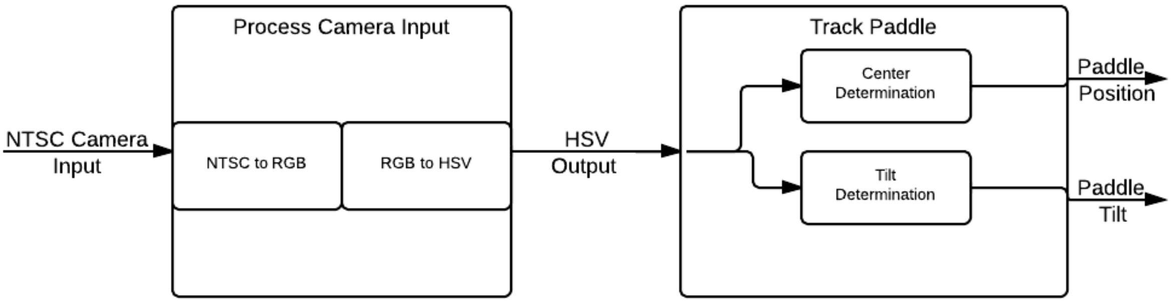

Figure 4: Video processing components. NTSC camera input is first decoded into an RGB stream. The RGB stream is converted to HSV and then sent for more processing. The paddle tracking component determines the center of mass of the paddle and its tilt.

We will test this module by displaying the calculated center of mass of the paddle on the screen against the camera's image. Similarly, the tilt will be visually confirmed. If the tilt of the paddle is not calculating correcting, tests with mock input coordinates may be used in the ModelSim test bench. We will also make use of a crosshairs module which will output the HSV value of a particular pixel on the screen. This will allow us to determine the necessary color range for paddle detection.

3.2 Level Creator Module

The level creator module will let the user arrange the bricks to use in the game. The user will use mouse input to position bricks. The user will be able to select from a variety of predesigned brick styles from which the user can select. A representative view of the level creator is shown below in Figure 5. The level creator module will be coded by Jonathan.

The level creator module will primarily be tested using visual checks to make sure the logic is functioning.

A stretch goal is to be able to save and load levels. Saving the level will require more testing. This will likely be tested using the logic analyzer to make sure everything is saved correctly. The load level part of the brick breaker logic module will also confirm if the blocks are saved correctly.

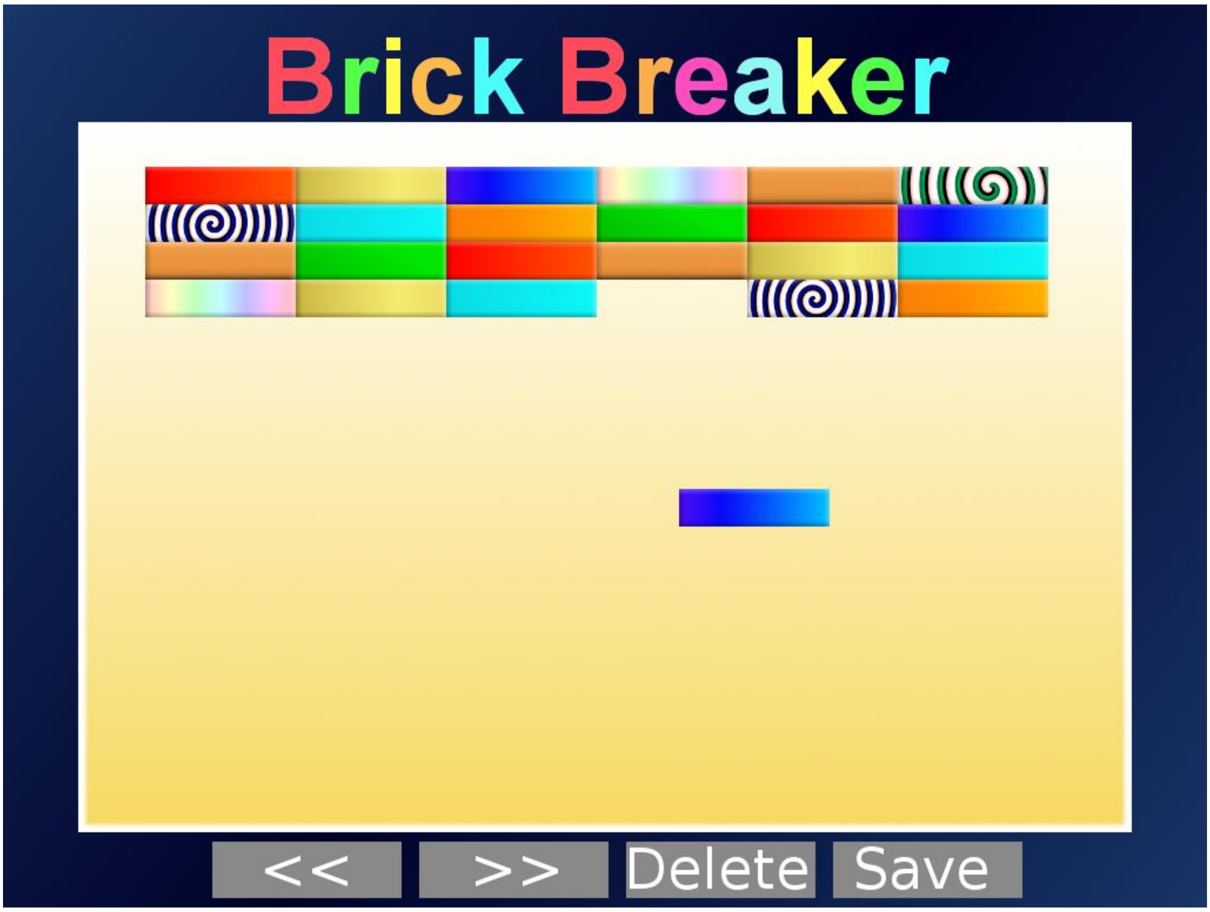

Figure 5: A representative view of the level creator. A user will move the block to place using the mouse. The block's style will be able to be changed by clicking on the cycle left and cycle right buttons.

3.3 Brick Breaker Logic Module

The brick breaker logic module handles the basic gameplay. This module will take in the coordinates of the Ping-Pong paddle as input and determine where the onscreen paddle should be positioned. This module will also place all the remaining bricks on the screen. The original placement of the bricks will be determined from the original level creator.

The brick breaker logic module also determines where the ball should be. This module tracks collisions of the ball with the paddle, bricks, and sides of the screen. If the ball reaches the bottom of the screen, the player will have "lost" a ball. The game will be over if all the balls are lost or if all the bricks are broken. The module will determine if bricks are broken by determining if the ball overlaps with a brick.

We will primarily test this module using the buttons on the 6.111 labkit to simulate the output from the camera. If needed, we will test this module by writing a Verilog test fixture and running a simulation of inputs (ball position, paddle position) and verifying whether the bricks properly disappear.

3.4 Graphics Module

The graphics module takes in a description of what should be shown on the game screen and outputs the corresponding VGA output for the monitor. The graphics module will be tested as we implement basic game features. The module involves visual data so verifying the screen output is correct should be sufficient for testing.

3.5 Sound Module

The sound module will generate different sounds related to events that occur in the game. The module will take in a signal corresponding to different collisions of the ball with the paddle, brick, and screen edge, and output specific sounds for each of these events. Ideally, the loudness of the sound will represent the speed of the collision.

We will test the sound module by writing a Verilog test fixture to test that the different sounds play. We will test that the sound module is properly integrated into the game through basic gameplay.

4 Proposed Project Plan

The modularity of this game allows us to each tackle different parts of the game's functionality in parallel. The plan is for Jemale to work on the tracking of the paddle using the camera and Jonathan to work on a level creation component that will tie in to the game logic. Towards the end, we will collaborate on adding on extra effects. This plan is shown below in Figure 6.

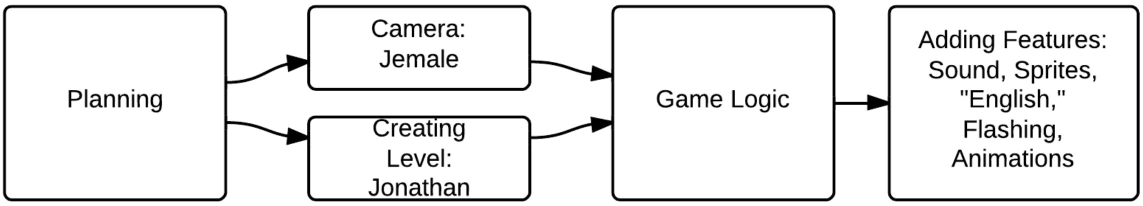
Planning
Camera: Jemale
Creating Level: Jonathan
Game Logic
Adding Features: Sound, Sprites, "English," Flashing, Animations

Figure 6: Planned timeline. After planning together, we decided that Jemale will code the camera tracking and Jonathan will code a level creation system. Afterwards, we will reconvene and collaborate on game logic and use the extra time to add more features.

We have set a few key dates to help us start on target for the above project plan. These key dates are given in the table below.

Week of 11/11	Continue implementing video modules Begin game logic module(s) Continue level creation modules
Week of 11/18	Begin implementing sound module Finish level creation module Finish video modules
Week of 11/25	Finish game logic module(s) Begin integrating modules Finish sound module
Week of 12/2	Testing and Debugging

There also are a number of future key submission dates. The submission key dates are given in this table below.

Date	Assignment
11/25	Project Status Update with mentor
12/09	Final Project Checkoff w/ staff 4-9pm
12/10	Project demos and videotaping 6-11pm
12/11	Final Project Report due (submit PDF by noon)

We look forward to making progress on Brick Breaker.

5 Appendix 1: Resources

Item	Cost	Status
Camera	Freely provided	Acquired
Mouse	Freely provided	Acquired
Ping-Pong paddle	$5.99	Purchased and arrived